

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-010796

(43)Date of publication of application : 14.01.2000

51)Int.Cl.

G06F 9/46  
G06F 11/34  
G06F 15/00

(21)Application number : 11-114162

(71)Applicant :

NIPPON TELEGR & TELEPH CORP <NTT>

(22)Date of filing : 21.04.1999

(72)Inventor :

SAKAMOTO YASUHISA  
KISHI KOJI  
WASHIMI TAKUYA

(30)Priority

Priority number : 10128262 Priority date : 22.04.1998 Priority country : JP

(54) METHOD FOR OBTAINING UTILIZING DATA OF APPLICATION ON INFORMATION PROCESSOR, ITS INFORMATION PROCESSOR AND STORAGE MEDIUM

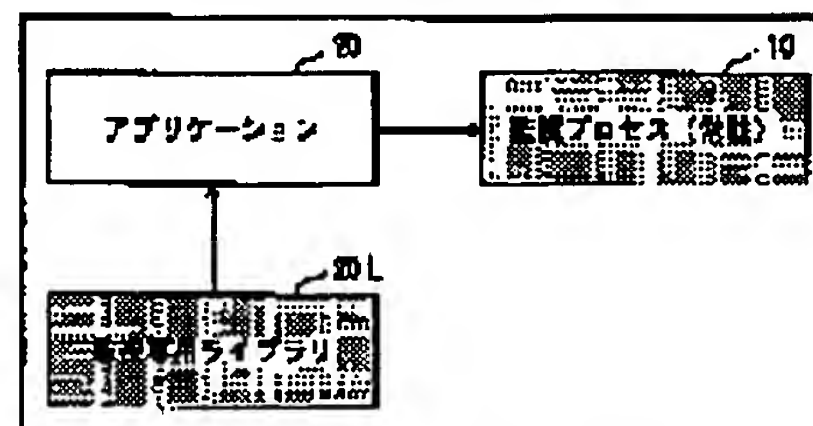
(57)Abstract:

PROBLEM TO BE SOLVED: To provide a method for efficiently obtaining sufficient utilizing data required for executing utilization trend investigation, effect measurement and advertizing medium value evaluation concerning various kinds of applications on an information processor.

SOLUTION: At the time of starting an application 20 being the monitoring object of a monitoring process 10 resident in the information processor, the process 10 inserts a library only for monitoring 20L, event information generated at the application 20 is obtained by the library 20L, and this obtained event information is transmitted to the process 10. Based on event information obtained by the process 10, the process 10 prepares utilization data and stores this prepared utilization data in a file.

101:情報処理装置

アプリケーション移動時 (初期設定)



## LEGAL STATUS

[Date of request for examination] 21.04.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3097033

[Date of registration] 11.08.2000

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C): 1998,2003 Japan Patent Office

BEST AVAILABLE COPY

(19)日本国特許庁（J P）(12)公開特許公報（A）(11)特許出願公開番号  
特開2000－10796  
（P2000－10796A）  
(43)公開日 平成12年1月14日(2000.1.14)

(51)Int.Cl.<sup>7</sup>識別記号F Iテマコード（参考）  
G 0 6 F 9/463 3 0G 0 6 F 9/463 3 0 C  
3 4 03 4 0 A  
11/3411/34C  
15/0015/003 1 0 A

審査請求 有 請求項の数 8 O L （全 11 頁）

(21)出願番号	特願平11-114162	(71)出願人	000004226 日本電信電話株式会社 東京都千代田区大手町二丁目3番1号
(22)出願日	平成11年4月21日(1999.4.21)	(72)発明者	坂本 泰久 東京都新宿区西新宿三丁目19番2号 日本 電信電話株式会社内
(31)優先権主張番号	特願平10-128262	(72)発明者	岸 晃司 東京都新宿区西新宿三丁目19番2号 日本 電信電話株式会社内
(32)優先日	平成10年4月22日(1998.4.22)	(72)発明者	鷺見 卓哉 東京都新宿区西新宿三丁目19番2号 日本 電信電話株式会社内
(33)優先権主張国	日本（J P）	(74)代理人	100070150 弁理士 伊東 忠彦

(54)【発明の名称】 情報処理装置上のアプリケーションの利用データ取得方法、その情報処理装置及び記録媒体

(57)【要約】

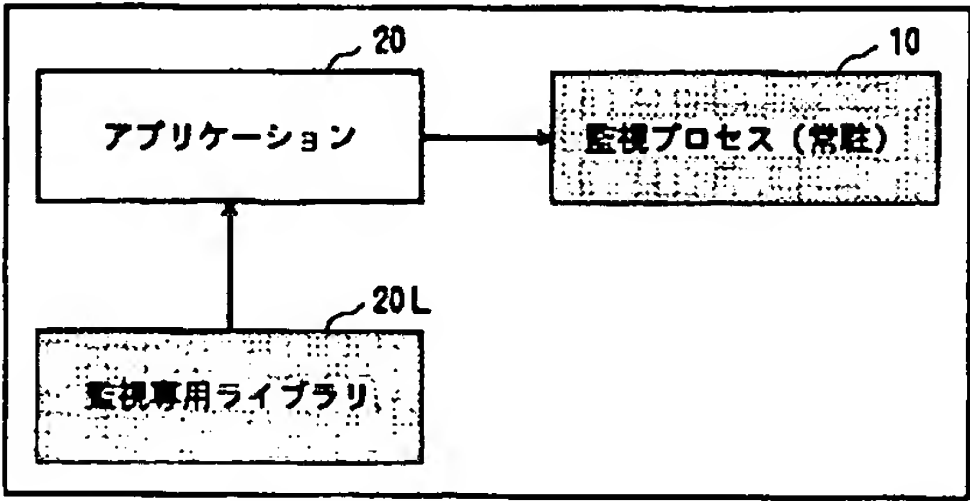
【課題】 情報処理装置上の多種多様なアプリケーションについて、利用動向調査、効果測定、広告媒体価値評価を行うに必要十分な利用データを効率的に取得する方法、情報処理装置及び記録媒体を提供することを目的とする。

【解決手段】 情報処理装置に常駐する監視プロセスの監視対象であるアプリケーションの起動時に、上記監視プロセスが監視専用ライブラリを挿入し、アプリケーションで発生したイベント情報を、上記監視専用ライブラリが取得し、この取得したイベント情報を上記監視プロセスに伝達し、上記監視プロセスが取得した上記イベント情報に基づいて、上記監視プロセスが、利用データを作成し、この作成された利用データをファイルに保存する。

本発明の第1の実施例である情報処理装置101を示すブロック図であり、アプリケーション起動時（初期処理）の状態を示す図

101：情報処理装置

アプリケーション起動時（初期処理）



## 【特許請求の範囲】

【請求項1】 情報処理装置上のアプリケーションの利用データを取得する方法であって、

上記情報処理装置に常駐する監視プロセスの監視対象であるアプリケーションの起動時に、上記監視プロセスの命令に基き監視専用ライブラリが挿入される監視専用ライブラリ挿入段階と、

アプリケーションで発生したイベント情報を、上記監視専用ライブラリが取得し、この取得したイベント情報を上記監視プロセスに伝達するイベント情報伝達段階と、  
上記監視プロセスが取得した上記イベント情報に基づいて、上記監視プロセスが、利用データを作成し、この作成された利用データをファイルに保存する利用データ保存段階とを有することを特徴とする方法。

【請求項2】 上記イベント情報伝達段階は、アプリケーションとオペレーティングシステムとの間に、またはアプリケーションとライブラリとの間に上記監視専用ライブラリが割り込み、上記アプリケーション又はオペレーティングシステムで発生したイベント情報を途中で横取りする形で取得し、アプリケーション固有処理に基づいて、イベントの解釈と取捨選択とを行う段階であることを特徴とする請求項1に記載の方法。

【請求項3】 1つの上記監視プロセスが複数のアプリケーションを監視する場合、上記各アプリケーションに挿入された監視専用ライブラリは、上記アプリケーション又はオペレーティングシステムで発生した利用データを、上記1つの監視プロセスに集約するものであることを特徴とする請求項1に記載の方法。

【請求項4】 アプリケーションの利用データを取得する情報処理装置であって、  
上記情報処理装置に常駐する監視プロセスの監視対象であるアプリケーションの起動時に、上記監視プロセスの命令に基き監視専用ライブラリが挿入される監視専用ライブラリ挿入手段と、  
アプリケーション又はオペレーティングシステムで発生したイベント情報を、上記監視専用ライブラリが取得し、この取得したイベント情報を上記監視プロセスに伝達するイベント情報伝達手段と、  
上記監視プロセスが取得した上記イベント情報に基づいて、上記監視プロセスが、利用データを作成し、この作成された利用データをファイルに保存する利用データ保存手段とを有することを特徴とする情報処理装置。

【請求項5】 上記イベント情報伝達手段は、アプリケーションとオペレーティングシステムとの間に、またはアプリケーションとライブラリとの間に上記監視専用ライブラリが割り込み、上記アプリケーション又はオペレーティングシステムで発生したイベント情報を途中で横取りする形で取得し、アプリケーション固有処理に基づいて、イベントの解釈と取捨選択とを行う手段であることを特徴とする請求項4に記載の情報処理装置。

【請求項6】 1つの上記監視プロセスが複数のアプリケーションを監視する場合、上記各アプリケーションに挿入された監視専用ライブラリは、上記アプリケーション又はオペレーティングシステムで発生した利用データを、上記1つの監視プロセスに集約するものであることを特徴とする請求項4に記載の情報処理装置。

【請求項7】 情報処理装置上でアプリケーションの利用データを取得するプログラムを記録したコンピュータ読み取り可能な記録媒体であって、該プログラムは、  
上記情報処理装置に常駐する監視プロセスの監視対象であるアプリケーションの起動時に、上記監視プロセスの命令に基き監視専用ライブラリが挿入される監視専用ライブラリ挿入手順と、

アプリケーション又はオペレーティングシステムで発生したイベント情報を、上記監視専用ライブラリが取得し、この取得したイベント情報を上記監視プロセスに伝達するイベント情報伝達手順と、  
上記監視プロセスが取得した上記イベント情報に基づいて、上記監視プロセスが、利用データを作成し、この作成された利用データをファイルに保存する利用データ保存手順とをコンピュータに実行させることを特徴とする記録媒体。

【請求項8】 情報処理装置上でWWWブラウザの利用データを取得するプログラムを記録したコンピュータ読み取り可能な記録媒体であって、該プログラムは、  
利用者によって上記WWWブラウザが起動されると、監視プロセスの制御スレッドがこれを検知し、監視専用ライブラリをロードするように、上記WWWブラウザに制御スレッドが要求する手順と、

30 実行時メモリ空間上で、通信ライブラリの先頭部分のプログラムを、上記ロードされた上記監視専用ライブラリが書き換え、ブラウザ本体と上記通信ライブラリとの間に、上記監視専用ライブラリ自身を挿入する手順と、  
上記WWWブラウザ内の上記監視専用ライブラリからイベント情報を受け取るための監視スレッドを、上記制御スレッドが起動し、この起動された上記監視スレッドが、上記通信ライブラリとの接続を確立する手順と、  
利用者が目的のページを指定すると、通信ライブラリ内のネットワーク送信関数を用い、サーバに対して上記WWWブラウザがファイル要求を出し、上記監視専用ライブラリを経由してから上記通信ライブラリに、上記関数呼び出しイベントを送る手順と、  
関数呼び出しイベントまたは関数終了イベントの通過時に、ブラウザ固有情報に基づいて、上記通信ライブラリが、関数を取捨選択し、必要な引数情報を保持する手順と、

上記通信ライブラリが、保持した情報を上記監視スレッドに送る手順と、

40 得られた一次情報に基づいて、必要であれば、上記監視スレッドが、情報を追加・加工し、利用データを作成す

る手順と、

作成された利用データを、上記監視スレッドが書き込みスレッドに送る手順と、

得られた利用データを、上記書き込みスレッドが利用履歴ファイルに書き出す手順とをコンピュータに実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理装置上のアプリケーションの利用データの取得する方法に関し、特に、利用動向調査、インターネットメディアの効果測定・広告媒体価値評価技術に適用するために、例えばコンピュータ等の情報処理装置上のアプリケーションの利用データを取得する方法、装置及び記録媒体に関する。

【0002】

【従来の技術】近年、インターネットを中心として、WWWや電子メール等のネットワークサービスが一般家庭にも浸透している。これに伴って、ネットワーク通信機能付きパソコンの普及が進み、ネットワーク通信機能付きテレビ等一般家電製品とコンピュータとの融合が進んでいる。

【0003】情報端末からの要求に従って情報サーバがサービスを提供するサーバクライアント方式のネットワーク通信サービスにおいて、そのサービスの利用状況を取得しようとする場合、現在では、情報サーバ側で記録をとる方法が主流である。情報端末から情報サーバに要求がある度に、その提供サービスの内容、利用者情報、時刻等の利用データが、対応する情報サーバに記録される。

【0004】一方、情報端末側のオペレーティングシステムが、サービスの利用状況を記録する機能を有する場合もあるが、アプリケーション毎の利用データは、アプリケーション依存性が高いので、エラー情報等の極めて限られた情報しか、取得することかできない。図1は、情報端末において動作するアプリケーションが利用データを取得する場合における従来方式の説明図である。

【0005】情報端末において動作するアプリケーションが利用データを取得する場合、従来、図1(1)～(3)に示すプロセス間通信方式、ライブラリ置換方式、アプリケーション改造方式の3つの方式が採用されている。図1(1)に示すプロセス間通信方式は、オペレーティングシステムを介して複数の監視プロセス1同士がデータ通信を行うことによって、所定の監視プロセス1における所定のアプリケーション2の状態を、別の監視プロセス1が知ることができる方式である。

【0006】たとえば、マイクロソフト社Windows OSでは、OLE方式やDDE方式と呼ばれるプロセス間通信方式を実現することができ、アプリケーションとしてWWWブラウザを使用する場合、外部プロセスか

ら問い合わせを行なうことによって、表示中のページアドレス情報等を取得することができる。図1(2)に示すライブラリ置換方式は、アプリケーション2aから呼び出される関数群であるライブラリを、監視機能付きのライブラリ2aLと置き換え、この置き換えによって、アプリケーション2aからの関数呼び出しの履歴をライブラリ2aL側で知ることができる方式である。

【0007】たとえば、マイクロソフト社Windows OSのSocket32.dllという標準のネットワーク通信ライブラリを、監視機能を有する通信ライブラリと名前を同じにして置き換え、この置き換えによって、WWWブラウザからライブラリの各種関数を呼び出した履歴を取得することができる。図1(3)に示すアプリケーション改造方式は、改造したアプリケーション2bを使用することによって、アプリケーション2b自体に必要な利用データを利用履歴ファイル3に記録する方式である。

【0008】

【発明が解決しようとする課題】しかしながら、図1(1)に示す従来のプロセス間通信方式の場合、一般的に、利用履歴の記録や公開は利用者の直接の利益にならないので、アプリケーション本来の機能として実装されにくく、限定されたデータしか取得できない場合が多いという問題がある。

【0009】また、図1(3)に示す従来のアプリケーション改造方式や、図1(2)に示す従来のライブラリ置換方式の場合、アプリケーションまたはライブラリ毎に改造を加えなければならないので、汎用性がないという問題がある。また、アプリケーションまたはライブラリ毎に改造を加えた上で、再配布しなければならないので、大きなコストがかかる。特に最近では、アプリケーションのバージョンアップが頻繁に発生するので、上記の問題点が顕著になっている。

【0010】本発明は、上記の点に鑑みてなされたものであり、情報端末を含む情報処理装置上の多種多様なアプリケーションについて、利用動向調査、効果測定、広告媒体価値評価を行うに必要十分な利用データを効率的に取得することができる情報処理装置上のアプリケーションの利用データの取得方法、その情報処理装置及びそのプログラムを記録した記録媒体を提供することを目的とするものである。

【0011】なお、利用データを効率的に取得することとは、どのアプリケーションにおいても、アプリケーション自体を改造せずに利用データを取得することである。

【0012】

【課題を解決するための手段】上記の目的を達成するために、本発明は次のように構成される。本発明は、情報処理装置上のアプリケーションの利用データを取得する方法であって、上記情報処理装置に常駐する監視プロセ



スの監視対象であるアプリケーションの起動時に、上記監視プロセスの命令に基き監視専用ライブラリが挿入される監視専用ライブラリ挿入段階と、アプリケーションで発生したイベント情報を、上記監視専用ライブラリが取得し、この取得したイベント情報を上記監視プロセスに伝達するイベント情報伝達段階と、上記監視プロセスが取得した上記イベント情報に基づいて、上記監視プロセスが、利用データを作成し、この作成された利用データをファイルに保存する利用データ保存段階とを有する。

【0013】本発明によれば、アプリケーションに監視専用ライブラリを挿入し、イベント情報を横取りすることとしたので、アプリケーションを改造することなく、詳細な利用データを得ることができる。上記構成において、上記イベント情報伝達段階は、アプリケーションとオペレーティングシステムとの間に、またはアプリケーションとライブラリとの間に上記監視専用ライブラリが割り込み、上記アプリケーションで発生したイベント情報を途中で横取りする形で取得し、アプリケーション固有処理に基づいて、イベントの解釈と取捨選択とを行うようにしてもよい。

【0014】本発明によれば、アプリケーション固有処理を変更するだけで、アプリケーションを改造することなく、詳細な利用データを得ることができる。更に、上記構成において、1つの上記監視プロセスが複数のアプリケーションを監視する場合、上記各アプリケーションに挿入された監視専用ライブラリは、上記アプリケーションで発生した利用データを、上記1つの監視プロセスに集約するようにしてもよい。

【0015】本発明によれば、複数のアプリケーションの利用データを効率的に取得することができる。また、上記目的を達成するために本発明は次のように構成してもよい。本発明はアプリケーションの利用データを取得する情報処理装置であって、上記情報処理装置に常駐する監視プロセスの監視対象であるアプリケーションの起動時に、上記監視プロセスの命令に基き監視専用ライブラリが挿入される監視専用ライブラリ挿入手段と、アプリケーションで発生したイベント情報を、上記監視専用ライブラリが取得し、この取得したイベント情報を上記監視プロセスに伝達するイベント情報伝達手段と、上記監視プロセスが取得した上記イベント情報に基づいて、上記監視プロセスが、利用データを作成し、この作成された利用データをファイルに保存する利用データ保存手段とを有する。

【0016】更に、上記目的を達成するために本発明は次のように構成してもよい。本発明は情報処理装置上でアプリケーションの利用データを取得するプログラムを＊（イベント名）（意味）

WM\_COMMAND……………ブラウザのボタン押下

WM\_DDE\_EXECUTE……他のアプリケーションからの指示

＊記録したコンピュータ読み取り可能な記録媒体であって、該プログラムは、上記情報処理装置に常駐する監視プロセスの監視対象であるアプリケーションの起動時に、上記監視プロセスの命令に基き監視専用ライブラリが挿入される監視専用ライブラリ挿入手順と、アプリケーションで発生したイベント情報を、上記監視専用ライブラリが取得し、この取得したイベント情報を上記監視プロセスに伝達するイベント情報伝達手順と、上記監視プロセスが取得した上記イベント情報に基づいて、上記監視プロセスが、利用データを作成し、この作成された利用データをファイルに保存する利用データ保存手順とをコンピュータに実行させる。

【0017】これらの発明においても、上記と同様な効果を得ることができる。

【0018】

【発明の実施の形態】図2は、本発明の第1の実施例であるアプリケーションの利用データを取得する情報処理装置101を示すブロック図である。また、この図は、情報処理装置101におけるアプリケーション起動時の状態（初期処理）を示す。本実施例において、情報処理装置101は例えばパソコンPC1である。情報処理装置101におけるアプリケーションが利用データを取得する。情報処理装置101に常駐する監視プロセス10が、監視対象となるアプリケーション20の起動時に、監視専用ライブラリ20Lをロードさせる命令を出し、この命令に応じて、アプリケーション20は、監視専用ライブラリ20Lをロードする。

【0019】図3は、情報処理装置101におけるアプリケーション実行時の状態（監視状態）を示すブロック図である。監視専用ライブラリ20Lは、オペレーティングシステムOSで発生するイベントのうち利用データの収集に必要なイベント情報（提供サービスの内容、利用者情報、時刻等の情報）を、アプリケーション20とオペレーティングシステムOSとの間に、割り込んで、途中で（アプリケーション20とオペレーティングシステムOSとの間で）横取りする形で取得し、後述のアプリケーション固有処理に基づいて、イベントを取捨選択し、この選択したイベントを解釈した後、解釈されたイベントを逐次監視プロセス10に伝達する。監視プロセス10は、取得したイベント情報に基づいて、利用データを作成し、この作成された利用データを利用履歴ファイル30に保存する。

【0020】ここで、アプリケーション固有処理について説明する。監視専用ライブラリ20Lが取得できる1次情報（イベント情報）には、様々な種類がある。たとえば、WWWブラウザアプリケーションである場合、OS-ブラウザ間のイベントとして、例えば、

WM\_KEYDOWN.....キーボードの入力

WM\_SETTEXT.....ファイルアドレス等文字列の変更

等の多数の情報を得ることができる。

【0021】しかし、利用データになるのは、この中の一部分であるので、これを選択するための情報が必要になり、また、その情報を、利用データとして必要なデータに変換する処理が必要になる。これを、アプリケーション固有処理という。たとえば、上記のうち、ファイルアドレスだけが利用データとして必要である場合、「多数のイベントからWM\_SETTEXTイベントだけを抽出し、その引数を取り出し、ファイルアドレスとして記録する」という手続きが、アプリケーション固有処理である。

【0022】なお、アプリケーション20とオペレーティングシステムOSとの間に割り込んでイベント情報を取得する（第1の方法と称する）代わりに、アプリケーション20とライブラリ25との間に割り込んで、途中で横取りする形でイベント情報を取得する（第2の方法と称する）ようにしてもよい。これらの方法について図4を用いて説明する。

【0023】第1の方法は、ステップA1とA2で示され、ステップA1において、オペレーティングシステムOSが発するイベント情報を監視専用ライブラリ20Lが取得し、その後、ステップA2として、アプリケーション\*

send.....相手先コンピュータにデータを送る

recv.....相手先コンピュータからデータを受け取る

gethostbyname...相手先コンピュータを見つける

connect.....接続状態をつくる

等、約50種類が含まれている。

【0026】図5は、本発明の第2の実施例である情報処理装置102を示すブロック図であり、複数のアプリケーションの実行時の状態（監視状態）を示す図である。情報処理装置102は、例えば、パソコンPC2で構成される。同図に示すように、1つの監視プロセス10が複数のアプリケーション21、22、23を監視する場合、各アプリケーション21～23に挿入されている監視専用ライブラリ21L、22L、23Lは、アプリケーション21～23で発生した利用データを、1つの監視プロセス10に集約する。

【0027】次に、アプリケーションとしてWWWブラウザを使用する場合を例にとって、上記実施例について説明する。WWWブラウザは、インターネット上のWWWサーバに蓄積されている情報を、ネットワークを通じて取り寄せ、この取り寄せた情報を閲覧するアプリケーションである。この場合、閲覧する情報の単位は、ページであり、「ページ」とは、主としてHTML言語によって記述された1つ以上の要素からなる情報の組である。上記情報の組を構成する各要素には、URLと呼ばれる情報空間上の位置を示すアドレスが付されている。

【0028】利用者は、WWWブラウザを用い、URL

\* ヨン20にそのイベント情報を渡す。第2の方法は、ステップB1とステップB2で示される方法である。ライブラリ25中のsend関数等の関数の先頭数バイトを書き換えることにより（ステップB1）、監視専用ライブラリに処理をジャンプさせて（ステップB2）、send関数等の関数の引数を監視専用ライブラリが取得する。

10 【0024】また、図3に示すライブラリ25は、アプリケーションの動作に必要な様々の機能を関数として共通化し、アプリケーションとは別のファイルとしたものである。特に機能に限定はない。ファイルの読み書き、ネットワークを介した送受信、様々の画面表示等基本的な処理は、その全てがライブラリ25で実現されている。

20 【0025】実施例で対象としているのは、wsck32.dllというネットワーク通信用ライブラリである。このライブラリにはインターネット上のデータ通信についての関数がまとめられている。電子メール、WWWブラウザ、ファイル転送等、殆どのインターネットアプリケーションは、このライブラリを用いる。関数としては、

30 を使用し、目的のページを指定することによって情報を見ることができる。また、ハイパーリンクによってページ同士を関係づけることができ、ページ上に表示されたハイパーリンクを利用者が押すことによって、関連する情報を次々と辿ることができる。WWWは、サーバクライアント方式のサービスであり、ページ参照時の基本的な動作は次のようなものである。

【0029】（1）クライアントであるブラウザからサーバに、要求（リクエスト）が送られる。

（2）サーバは、クライアントからの要求を受け取り、この要求に対応するページを応答（レスポンス）として送り返す。

40 （3）ブラウザは、応答を受信し、画面上に表示する。この一連の送受信手順がHTTPプロトコルとして定義されている。

【0030】図6は、上記各実施例における利用データの例を示す図であり、WWWブラウザの利用データとして役立つ項目を示す図である。図6（A）は、プロセス間通信方式でも取得できる情報であるが、図6（B）～（D）は、取得できない情報である。上記実施例によれば、監視専用ライブラリ21L～23Lがアプリケーション21～23にロードされ、OSからWWWブラウザ

に伝えられる各種入力イベントを、監視専用ライブラリ20L~23Lが監視することによって、図6(B)、(C)に示すような入力操作に起因する情報を得ることができる。

【0031】なお、「入力イベント」は、ボタンのマウスクリック、入力フィールドへの文字入力、メニューバーの使用等である。また、アプリケーション21~23にロードされた監視専用ライブラリ21L~23Lが、WWWブラウザ本体と通信用ライブラリとの間でやりとりされるネットワーク送受信に関する関数呼び出しイベントを監視することによって、図6(D)のようなアプリケーション21~23の内部の動作情報を得ることができる。

【0032】次に、上記実施例において、図6(D)に示すブラウザがサーバにファイルのアドレス(URL)を要求し、この要求されたファイルのアドレス(URL)を、関数呼び出しの監視によって得る動作を具体的に説明する図7は、本発明の第3の実施例である情報処理装置103を示すブロック図であり、WWWブラウザによる要求ファイル履歴を監視する動作を示す図である。

【0033】情報処理装置103は、例えばパソコンPC3で構成される。同図に示すように、監視プロセス10と、WWWブラウザ50と、利用履歴ファイル30とを有する。WWWブラウザ50は、ブラウザ本体51と、通信用ライブラリ52と、監視専用ライブラリ20Lとを有する。図8は、情報処理装置103において、WWWブラウザによる要求ファイル履歴を監視する動作を示すフローチャートである。図8を用いて動作を説明する。

【0034】情報処理装置103に常駐している監視プロセス10は、WWWブラウザ50が起動するまで待機している。そして、利用者によってWWWブラウザ50が起動される(S1)と、通信用ライブラリ52(たとえばマイクロソフト社Windows OSの場合Wsock32.dll)を含むライブラリ群が順次ロードされる。監視プロセス10の制御スレッド11は、WWWブラウザ50が起動したことを知ると(S2)、監視専用ライブラリ20Lをロードするように、WWWブラウザ50に要求する(S3)。

【0035】そして、ロードされた監視専用ライブラリ20Lは、実行時メモリ空間上で、通信用ライブラリ52の先頭部分のプログラムを書き換え、ブラウザ本体51と通信用ライブラリ52との間に、監視専用ライブラリ20L自身を挿入する(S4)。制御スレッド11が、WWWブラウザ50内の監視専用ライブラリ20Lからイベント情報を受け取るための監視スレッド12を起動する。

【0036】そして、利用者が、目的のページを指定すると、WWWブラウザ50は、通信用ライブラリ内のネッ

トワーク送信関数(通常はsend関数)を用い、外部にあるサーバに対してファイル要求を出す。この関数呼び出しイベントは、監視専用ライブラリ20Lを経由してから通信用ライブラリ52に送られる(S5)。具体的には、通信用ライブラリ52での関数処理が終わると、関数終了イベントが監視専用ライブラリ20Lを経由し、ブラウザ本体51に伝わる。

【0037】そして、監視ライブラリ20Lは、関数呼び出しイベントまたは関数終了イベントの通過時に、ブラウザ固有情報に基づいて関数を取捨選択し、必要な引数情報を保持する(S6)。要求ファイルアドレスを取得する場合には、ネットワーク送信関数(通常はsend関数)だけを対象とし、その引数に含まれるプロトコル名(たとえばhttp)、ホスト名(たとえばwww.company.com)、ファイル名(たとえばindex.html)を保持する。

【0038】そして、監視ライブラリ20Lは、保持した情報を監視スレッド12に送る(S7)。具体的には、監視スレッド12は、得られた一次情報に基づいて、必要であれば情報を追加・加工し、利用データを作成する。この場合、得られたプロトコル名、ホスト名、ファイル名を、URL形式(例・http://www.company.com/index.html)に変換し、ウィンドウ識別子情報と要求時刻情報とを追加する。監視スレッド12は、作成した利用データを書込みスレッド13に送り(S8)、書き込みスレッド13は、得られた利用データを、利用履歴ファイル30に書き出す(S9)。

【0039】図9は、本発明の第4の実施例である情報処理装置104を示すブロック図であり、WWWブラウザ50の要求ファイル履歴を監視する動作を示す図であり、1つの情報端末内で複数のWWWブラウザ50が実行された場合の動作を説明するブロック図である。情報処理装置104は、例えばパソコンPC4によって構成される。同図に示すように情報処理装置104は、監視プロセス10aと、利用履歴ファイル30と、WWWブラウザ50a、50b、50cと、オペレーティングシステムOSとを有するものである。監視プロセス10aは、監視スレッド12a、12b、12cと、書き込みスレッド13とを有する。WWWブラウザ50a、50b、50cは、それぞれ、監視専用ライブラリ50aL、50bL、50cLを有する。

【0040】図10は、アプリケーションの利用データ取得装置104において、得られる利用データファイルの例を示す図である。複数のブラウザ50a~50cが1つの情報端末内で実行された場合には、各ブラウザ毎に、ステップS2~S5の処理が行われ、WWWブラウザ50a~50cの数と同じ数だけ監視スレッド12a~12cが起動される。それぞれの監視スレッド12a~12cは、1つの書き込みスレッド13に利用データを



伝達する。

【0041】上記各実施例によれば、アプリケーションに監視専用ライブラリを挿入し、イベント情報を横取りすることによって、アプリケーション固有処理を変更し、通信ライブラリ内の関数の先頭の数バイトを書き換えるだけで、アプリケーションを改造することなく、詳細な利用データを得ることができる。上記実施例におけるアプリケーション固有処理は、具体的には、(1) wsock32.dllに含まれる約50種類の関数のうちsend関数だけを選択する処理、(2) send関数の引数のうち、プロトコル名、ホスト名、ファイル名を保持する処理である。

【0042】上述した通り、本発明の情報処理装置は、これまでに説明した処理を行うプログラムを搭載したパソコンで実現可能である。更に、パソコン以外の一般的なコンピュータシステムでも本発明の情報処理装置を実現することができる。図11はそのようなコンピュータシステムのハードウェア構成を示す図である。同図に示すように、本コンピュータシステムは、プログラムによる処理を実行するCPU200、データやプログラムを格納するメモリ201、データやプログラムを格納する外部記憶装置202、ディスプレイ203、キーボード204、インターネット等のネットワークとの接続を可能とする通信処理装置205を有する。本コンピュータシステムにこれまでに説明した処理を行うプログラムを搭載することにより、本コンピュータシステムは本発明の情報処理装置として使用できる。

【0043】更に、本発明は、コンピュータシステム以外の情報処理装置でも適用可能であり、その情報処理装置の利用データを効率的に取得することができる。次に、本発明のコンピュータ読み取り可能な記録媒体について説明する。本発明のコンピュータ読み取り可能な記録媒体は、本実施例では次の通りである。すなわち、利用者によってWWWブラウザが起動されると、監視プロセスの制御スレッドがこれを検知し、監視専用ライブラリをロードするように、上記WWWブラウザに制御スレッドが要求する手順と、実行時メモリ空間上で、上記通信ライブラリの先頭部分のプログラムを、上記ロードされた上記監視専用ライブラリが書き換え、ブラウザ本体と上記通信ライブラリとの間に、上記監視専用ライブラリ自身を挿入する手順と、上記WWWブラウザ内の上記監視専用ライブラリからイベント情報を受け取るための監視スレッドを、上記制御スレッドが起動し、この起動された上記監視スレッドが、上記通信ライブラリとの接続を確立する手順と、利用者が目的のページを指定すると、通信ライブラリ内のネットワーク送信関数を用い、サーバに対して上記WWWブラウザがファイル要求を出し、上記監視専用ライブラリを経由してから上記通信ライブラリに、上記関数呼び出しイベントを送る手順と、関数呼び出しイベントまたは関数終了イベントの通過時

に、ブラウザ固有情報に基づいて、上記通信ライブラリが、関数を取捨選択し、必要な引数情報を保持する手順と、上記通信ライブラリが、保持した情報を上記監視スレッドに送る手順と、得られた一次情報に基づいて、必要であれば、上記監視スレッドが、情報を追加・加工し、利用データを作成する手順と、作成された利用データを、上記監視スレッドが書き込みスレッドに送る手順と、得られた利用データを、上記書き込みスレッドが利用履歴ファイルに書き出す手順とを有するプログラムを記録した記録媒体である。

【0044】本記録媒体に記録されたプログラムを例えば図11に示すような汎用のコンピュータシステムにローディングすることによりそのコンピュータシステムを本発明の情報処理装置として使用できる。本記録媒体は、CD-ROM、磁気ディスク、ハードディスク、及び電子メモリ等で実現可能である。更に、図11に示すメモリ201や外部記憶装置202も本発明の記録媒体の例である。

【0045】本発明は、上記実施例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは勿論である。例えば、上記実施例ではアプリケーションとしてWWWブラウザを例にとったが、その他、例えばワープロソフト及び表計算ソフト等にも本発明は適用可能である。また、コンピュータのOSの例としてWindowsを例にとったが、例えば、UNIX等にも勿論本発明は適用可能である。

【0046】

【発明の効果】本発明によれば、アプリケーションに監視専用ライブラリを挿入し、イベント情報を横取りすることによって、アプリケーション固有処理を変更するだけで、アプリケーションを改造することなく、詳細な利用データを得ることができるという効果を奏する。

【図面の簡単な説明】

【図1】情報端末において動作するアプリケーションが利用データを取得する場合における従来方式の説明図である。

【図2】本発明の第1の実施例である情報処理装置101を示すブロック図であり、アプリケーション起動時(初期処理)の状態を示す図である。

【図3】情報処理装置101におけるアプリケーション実行時(監視状態)の状態を示すブロック図である。

【図4】監視専用ライブラリが割り込む方法を説明するための図である。

【図5】本発明の第2の実施例である情報処理装置102を示すブロック図である。

【図6】上記各実施例における利用データの例を示す図であり、WWWブラウザの利用データとして役立つ項目を示す図である。

【図7】本発明の第3の実施例である情報処理装置103を示すブロック図であり、WWWブラウザによる要求



ファイル履歴を監視する動作を示す図である

【図8】情報処理装置103において、WWWブラウザによる要求ファイル履歴を監視する動作を示すフローチャートである。

【図9】本発明の第4の実施例である情報処理装置104を示すブロック図である。

【図10】情報処理装置104において、得られる利用データファイルの例を示す図である。

【図11】コンピュータシステムの構成を示すブロック図である。

【符号の説明】

- 1 監視プロセス
- 2 アプリケーション
- 3 利用履歴ファイル
- 101～104 情報処理装置
- 10、10a 監視プロセス
- 11 制御スレッド

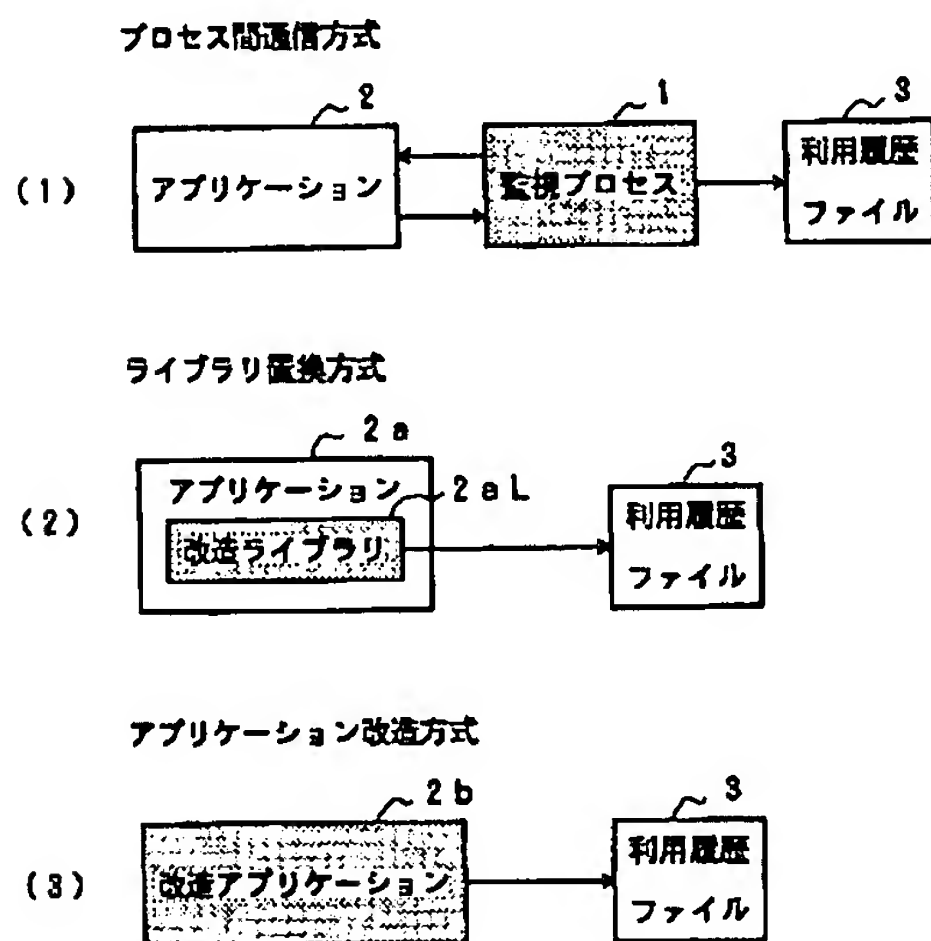
\*

- \* 12 監視スレッド
- 13 書き込みスレッド
- 20～23 アプリケーション
- 20L～23L 監視専用ライブラリ
- 25 ライブラリ
- 30 利用履歴ファイル
- 40 入力デバイス
- 50 WWWブラウザ
- 51 ブラウザ本体
- 10 52 通信ライブラリ
- 200 CPU
- 201 メモリ
- 202 外部記憶装置
- 203 ディスプレイ
- 204 キーボード
- 205 通信処理装置

\*

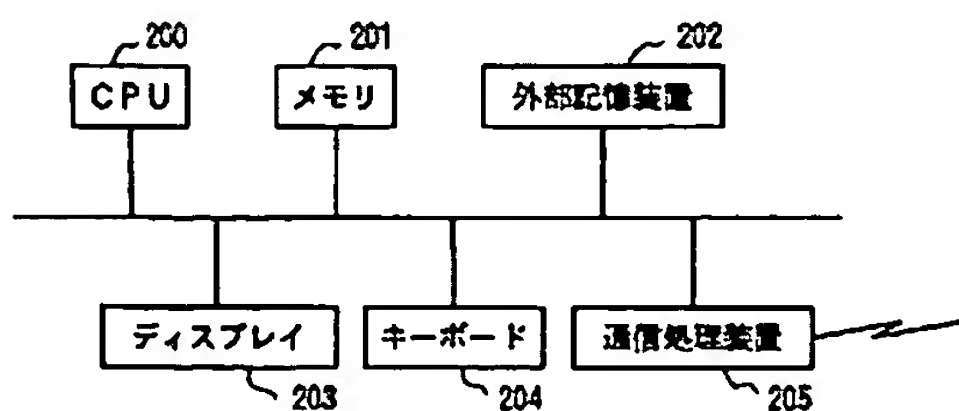
【図1】

情報端末において動作するアプリケーションが  
利用データを取得する場合における従来方式の説明図



【図11】

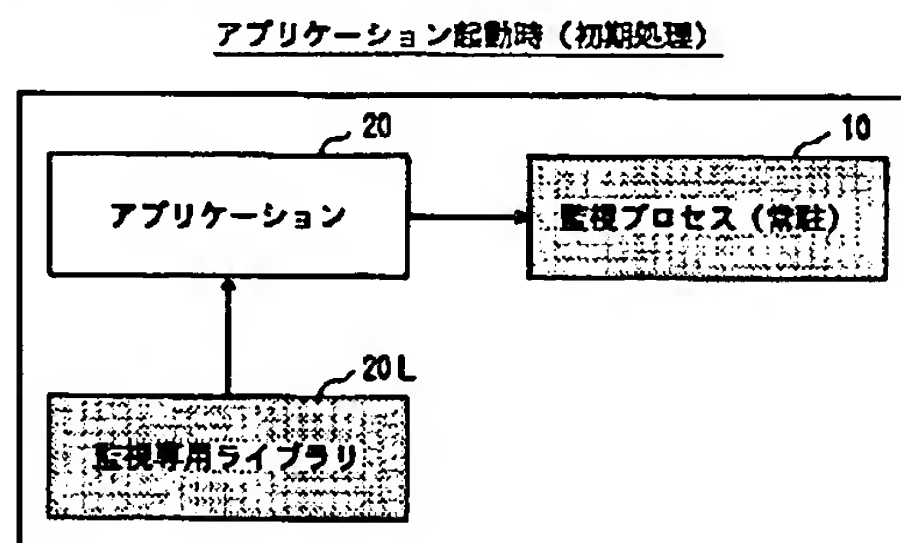
コンピュータシステムの構成を示すブロック図



【図2】

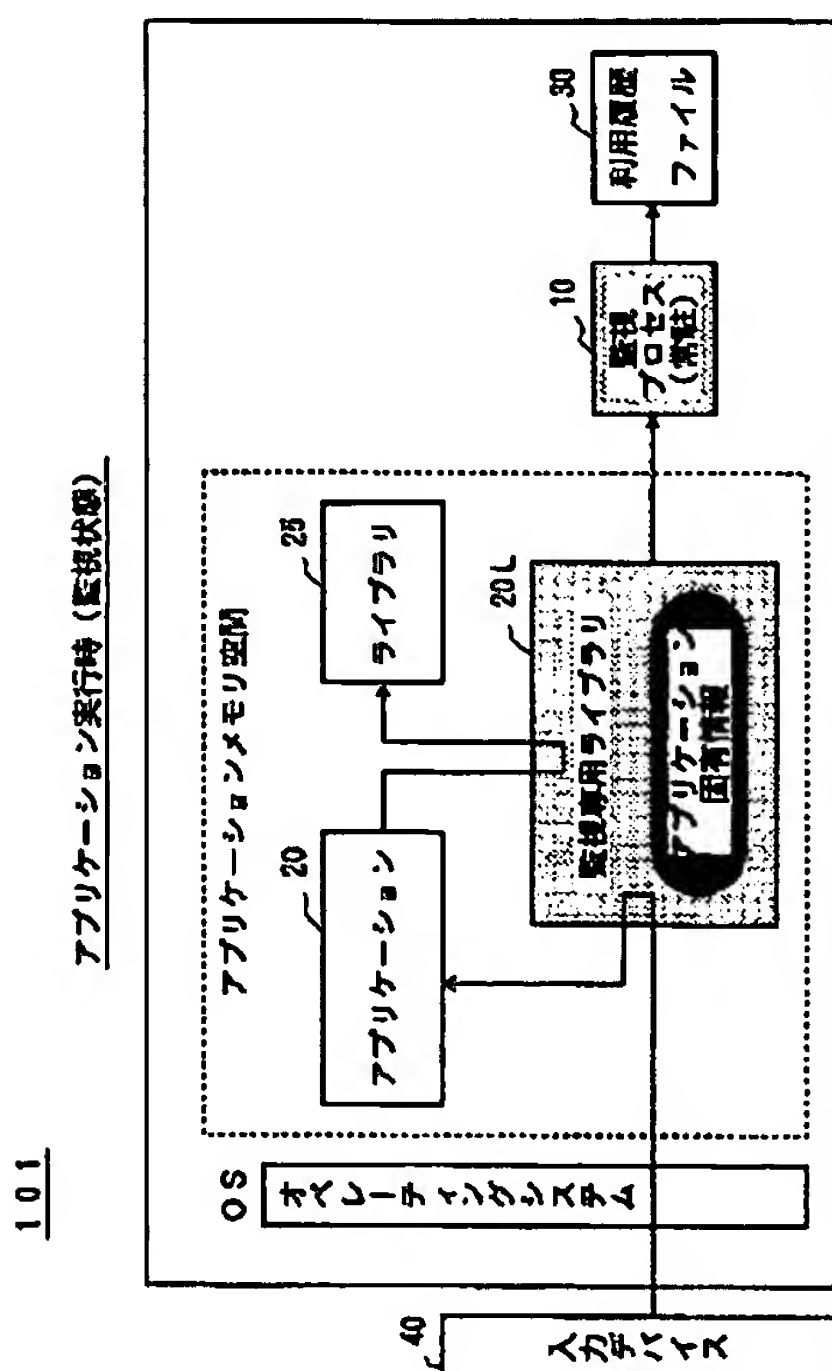
本発明の第1の実施例である情報処理装置101  
を示すブロック図であり、アプリケーション起動時  
(初期処理)の状態を示す図

101: 情報処理装置



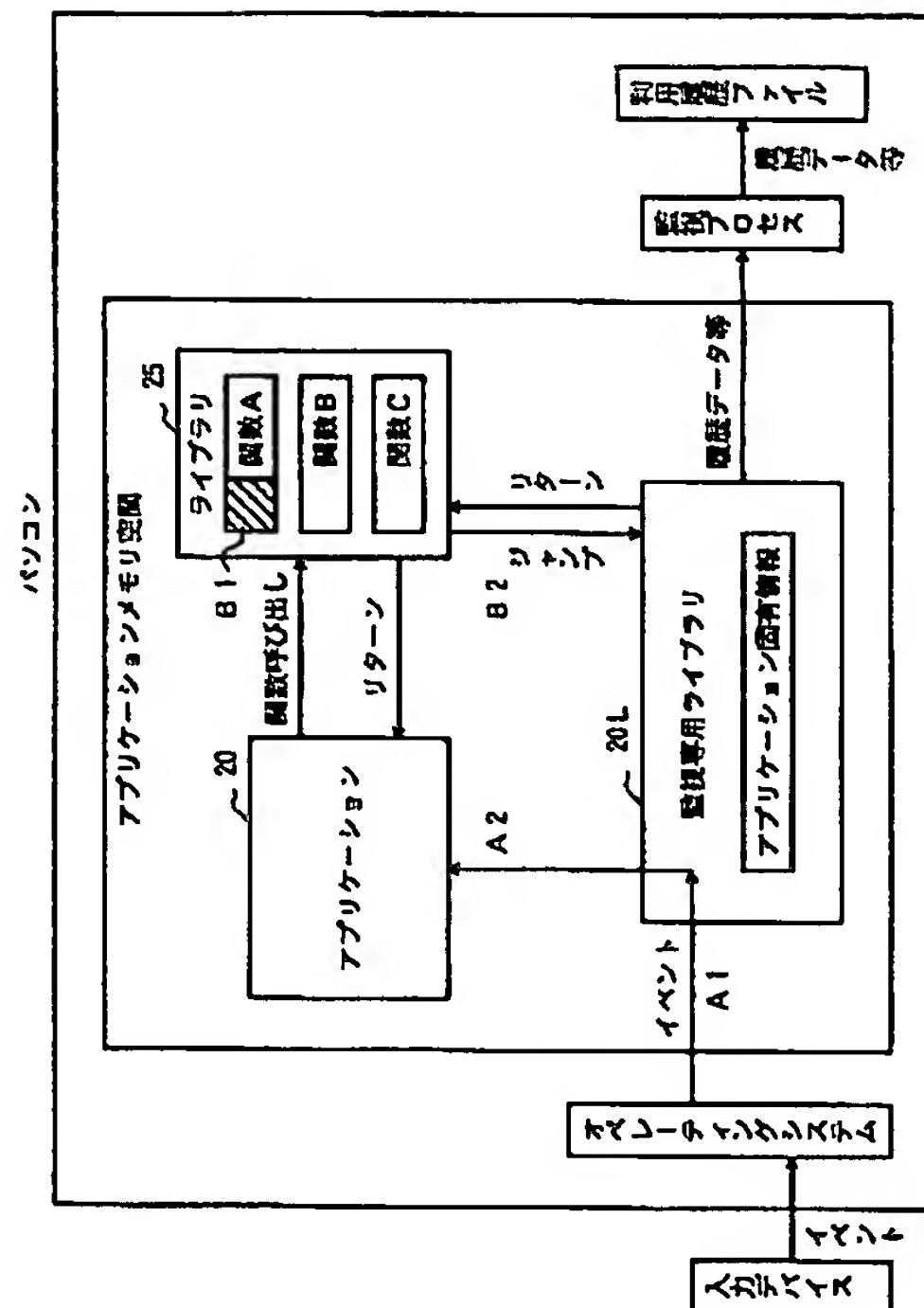
【圖3】

情報処理装置 101 におけるアプリケーション実行時  
(監視状態) の状態を示すブロック図



【図4】

監視専用ライブラリが割り込む方法を説明するための図



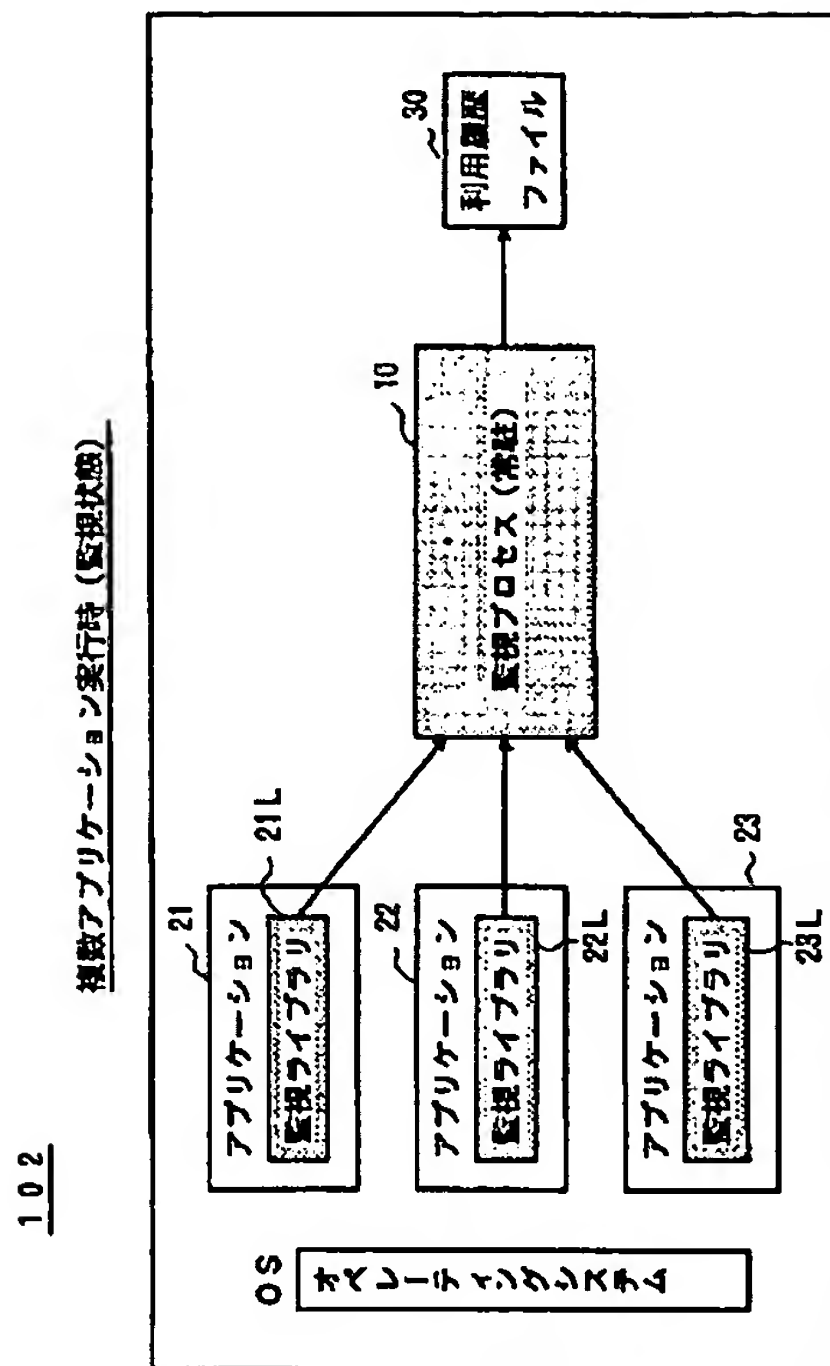
【図6】

各実施例における利用データの例を示す図であり、  
WWWブラウザの利用データとして役立つ項目を示す図

項目	例
(A) ブラウザ上に表示されたページの履歴	各ページのアドレス(URL)、タイトル、表示開始時刻、参照時間
(B) ページの指定方法、ページ間の移動方法、どのような操作によって目的のページに到達したか	ハイパーリンク、ブックマークの使用、アドレスのキー入力、進むボタン、戻るボタン、ブラウザ固有の特別メニュー、他のアプリケーションからの指示の跡
(C) ブラウザ上に表示されたページへの操作	ブックマークへの登録、ページの印刷
(D) ページに含まれる付帯要素のアドレス、すなわちブラウザがサーバに要求したファイルのアドレス	ページ内のイメージ画像のURL、ページ内のフレーム(ページを分割した枠)のURL

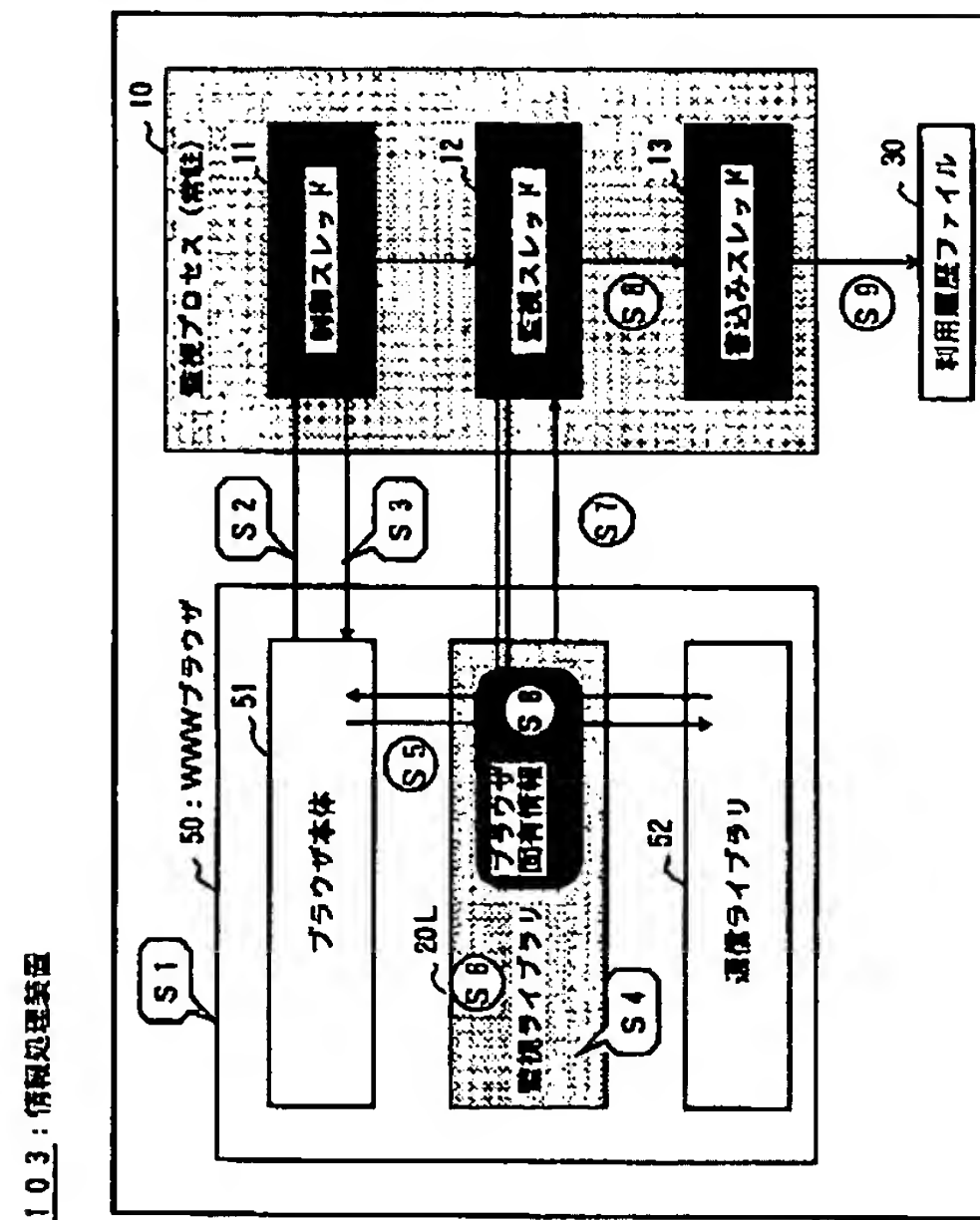
【図5】

本発明の第２の実施例である  
情報処理装置１０２を示すブロック図



【圖 7】

本発明の第３の実施例である情報処理装置１０３を示すブロック図であり、WWWブラウザによる要求ファイル履歴を監視する動作を示す図



【図 10】

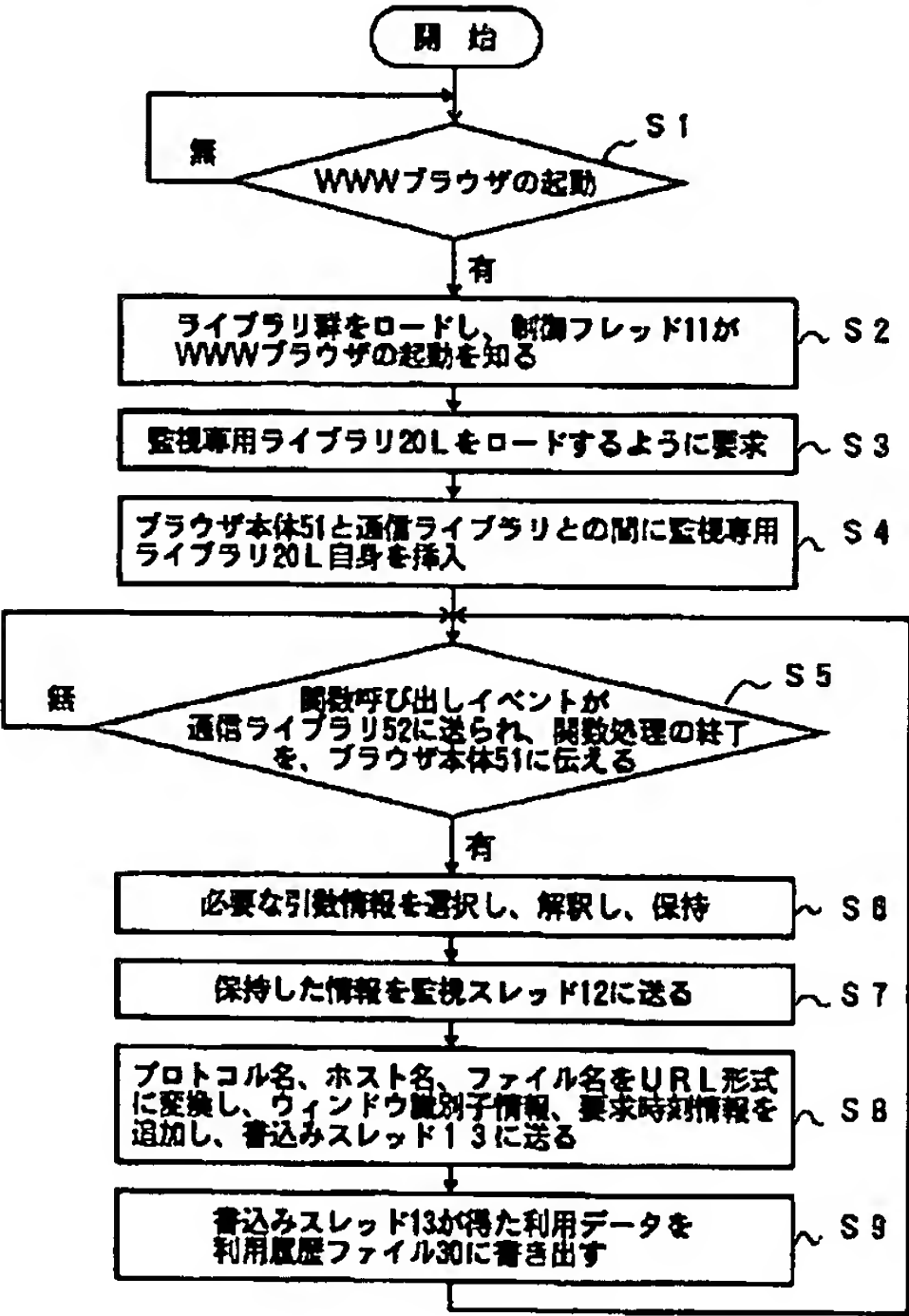
情報処理装置 104 において、  
得られる利用データファイルの例を示す図

ウィンドウ 識別子	要求時刻	ファイルアドレス (URL)
0001	1998/02/15 10:31:22	http://www.company.com/index.html
0001	1998/02/15 10:31:27	http://www.company.com/frame.html
0001	1998/02/15 10:31:35	http://www.company.com/gifs/inego.gif
0002	1998/02/15 10:32:08	http://www.company.com/index2.html
0002	1998/02/15 10:32:11	http://www.company.com/gifs/inego2.gif



【図8】

情報処理装置103において、WWWブラウザによる  
要求ファイル履歴を監視する動作を示すフローチャート



【図9】

本発明の第4の実施例である  
情報処理装置104を示すブロック図

